Block-Adaptive Parallel Explicit/Implicit MHD Simulations in Space Physics: the Art of Compromise

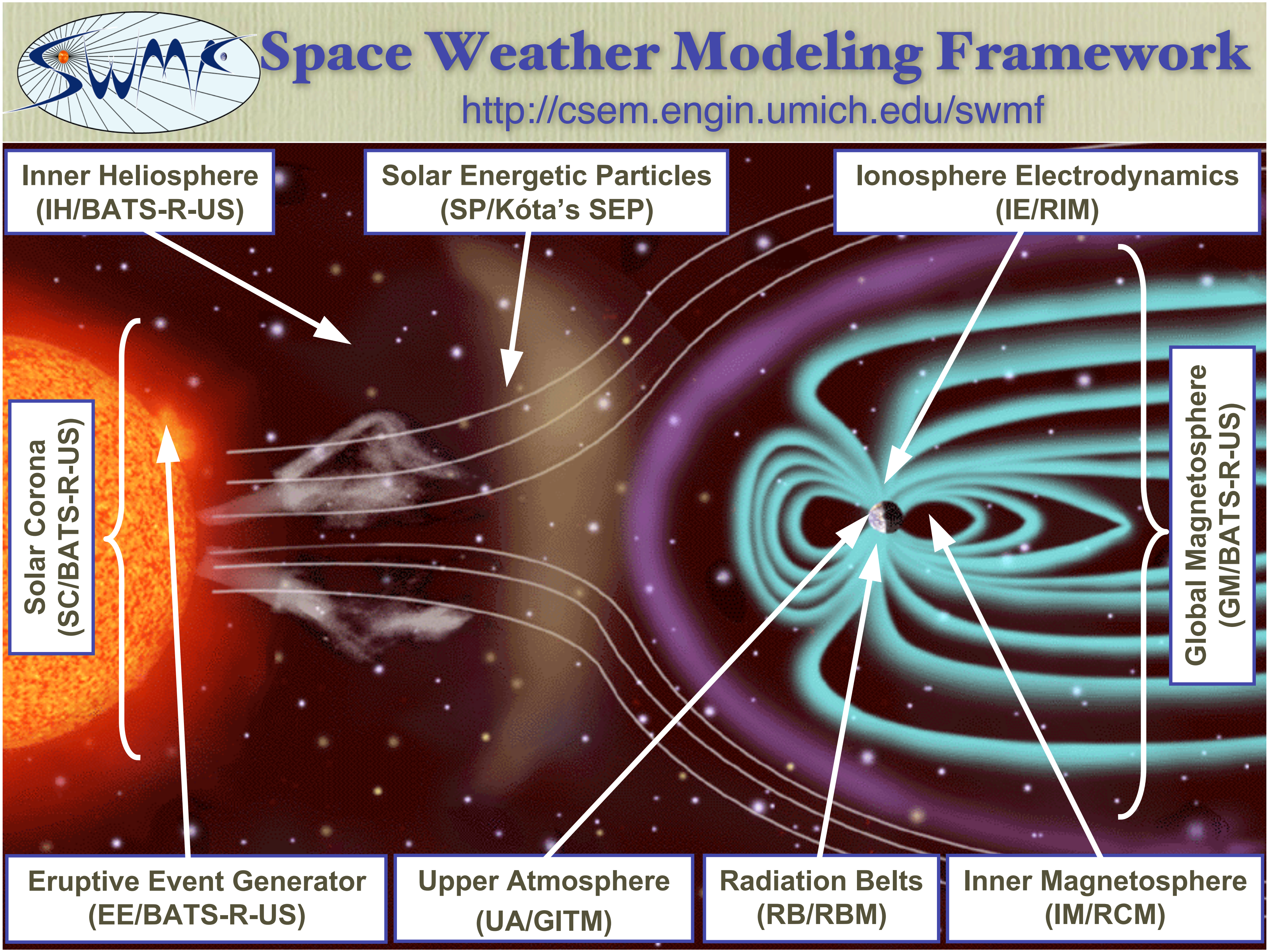
Gábor Tóth
Darren De Zeeuw, Kenneth Powell and Tamas Gombosi

Center for Space Environment Modeling
University of Michigan

Grants: NASA CT and DoD MURI

Outline of Talk

- Space Physics Applications: disparate scales
- Spatial Discretization: block adaptive grid
- Implicit Time Discretization: Jacobian free NKS
- Explicit/Implicit Scheme
- Numerical Tests
- Concluding Remarks


Space Weather Modeling Framework
http://csem.engin.umich.edu/swmf
Inner Heliosphere (IH/BATS-R-US)
Solar Energetic Particles (SP/Kóta's SEP)
Ionosphere Electrodynamics (IE/RIM)
Solar Corona (SC/BATS-R-US)
Global Magnetosphere (GM/BATS-R-US)
Eruptive Event Generator (EE/BATS-R-US)
Upper Atmosphere (UA/GITM)
Radiation Belts (RB/RBM)
Inner Magnetosphere (IM/RCM)

Vastly Disparate Scales

- **Spatial:**

  Resolution needed at Earth: $1/4\ R_E$

  Resolution needed at Sun: $1/32\ R_S$

  Sun-Earth distance: 1AU

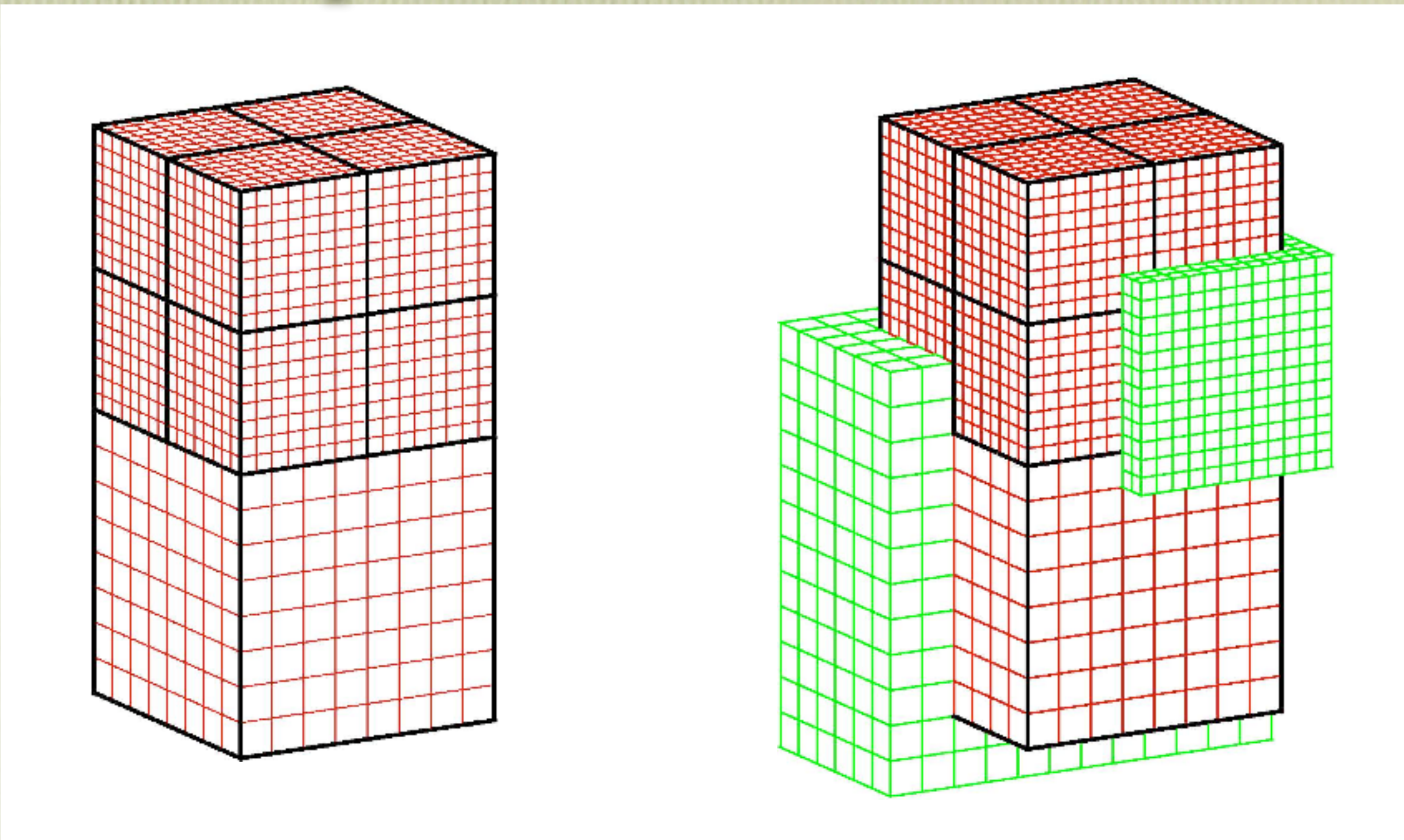
  $1\ AU = 215\ R_S = 23{,}456\ R_E$

- **Temporal:**

  CME needs 3 days to arrive at Earth.

  Time step is limited to a fraction of a second in some regions.

Adaptive Block Structure

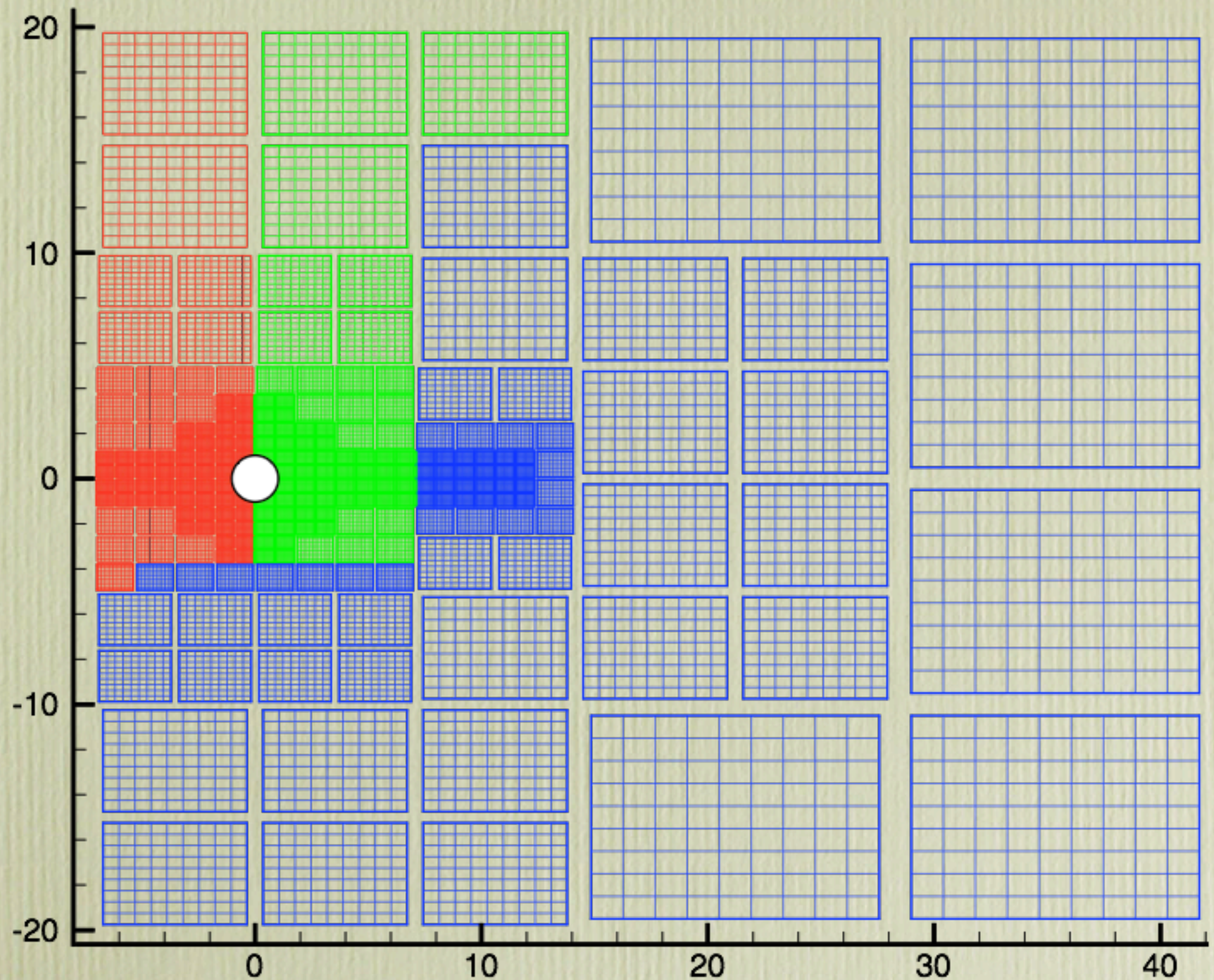


Each block is NxNxN

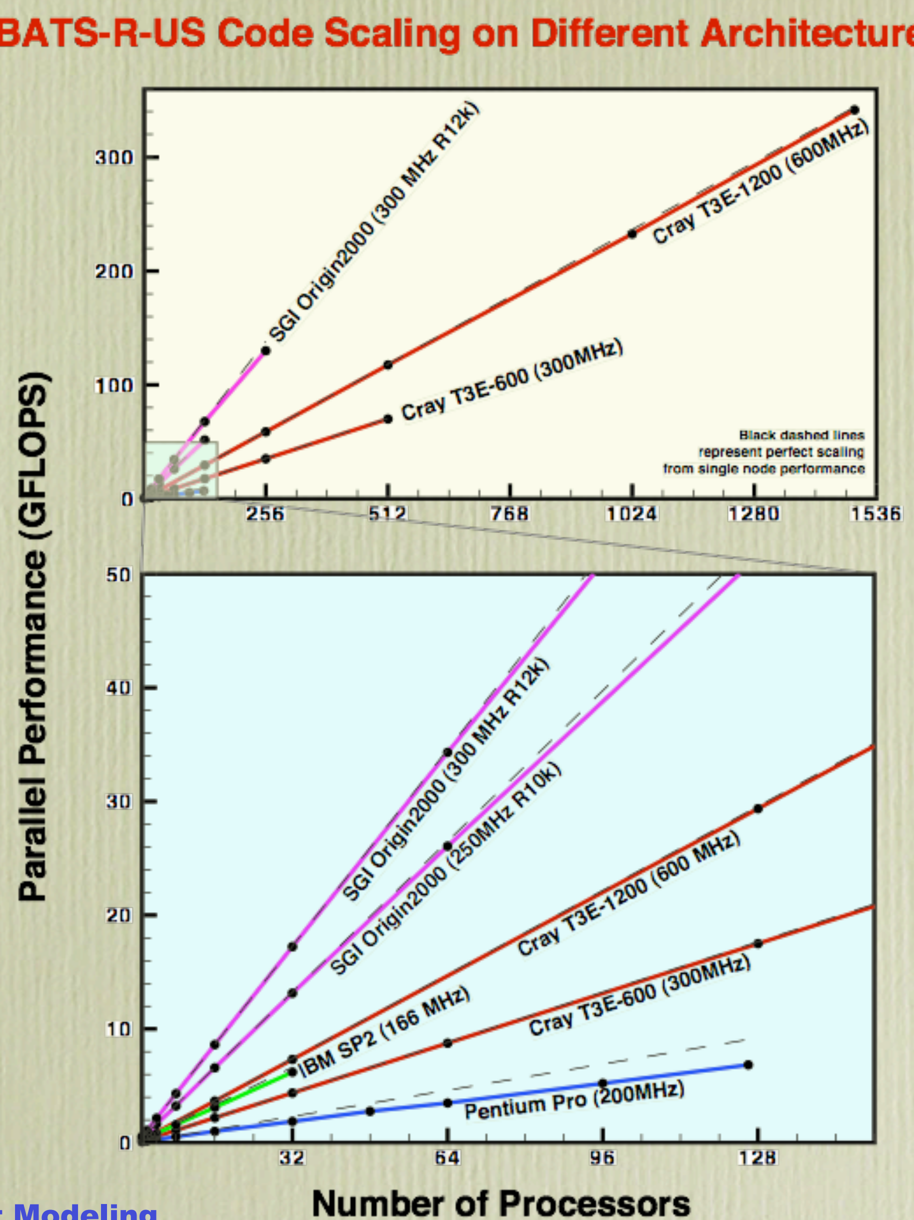
Blocks communicate with neighbors through "ghost" cells

Optimized Load Balancing:

based on Peano-Hilbert Space Filling Curve



Parallel Performance of Explicit Scheme



Why Implicit Time-Stepping Is Necessary?

- Explicit schemes have time step limited by CFL condition: $\Delta t < \Delta x$/fastest wave speed.
- The problem is particularly acute near planets with strong magnetic fields.
- High Alfvén speeds and/or small cells lead to much smaller time steps than required for accuracy: factor of 100 or even more.
- Implicit schemes do not have Δt limited by CFL.

Implicit Scheme

- Solve the non-linear semi-discretized PDE:

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}(\mathbf{U})$$

- Three-level second-order scheme (BDF2):

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} + \Delta t_n \left[\beta \mathbf{R}(\mathbf{U}^{n+1}) + (1-\beta) \frac{\mathbf{U}^{n} - \mathbf{U}^{n-1}}{\Delta t_{n-1}} \right]$$

where $\beta = (\Delta t_n + \Delta t_{n-1})/(2\Delta t_n + \Delta t_{n-1})$

- Use two-level scheme when U^{n-1} is not available/reliable.

Newton Linearization

- Linearize the non-linear term in the system of equations:

$$\mathbf{R}(\mathbf{U}^{n+1}) = \mathbf{R}(\mathbf{U}^{n}) + \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \cdot (\mathbf{U}^{n+1} - \mathbf{U}^{n}) + \mathcal{O}(\Delta t^2)$$

substitute back and rearrange:

$$\left[I - \Delta t_n \beta \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \cdot (\mathbf{U}^{n+1} - \mathbf{U}^{n}) = \Delta t_n \left[\beta \mathbf{R}(\mathbf{U}^{n}) + (1-\beta)\frac{\mathbf{U}^{n} - \mathbf{U}^{n-1}}{\Delta t_{n-1}}\right]$$

- Solving this linearized equation is equivalent with a single Newton iteration. Both the non-linear and the linear systems are second order accurate in time.

- Use spatially first order scheme for $\partial \mathbf{R}/\partial \mathbf{U}$ The scheme is still 2nd order accurate in space and time. Using the upwind scheme helps with diagonal dominance.

Krylov Solver

- Use GMRES (no restart)

BiCGStab

requires less memory but it is less robust

- Jacobian-free evaluation of matrix-vector products:

Iterations are

$$\left[I - \Delta t_n \beta \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \Delta\mathbf{U} = \Delta\mathbf{U} - \Delta t_n \beta \frac{\mathbf{R}(\mathbf{U}^n + \epsilon \Delta\mathbf{U}) - \mathbf{R}(\mathbf{U}^n)}{\epsilon} + \mathcal{O}(\epsilon)$$

- Krylov iterations are stopped when the initial error reduces by 10^3

A stricter tolerance does not improve the overall accuracy.

Variables must be normalized to make the errors comparable.

Schwarz Preconditioner

- Block by block preconditioning:
  Natural choice for block adaptive grid
  Simple matrix structure for Jacobian
  Results are independent of the number of processors.

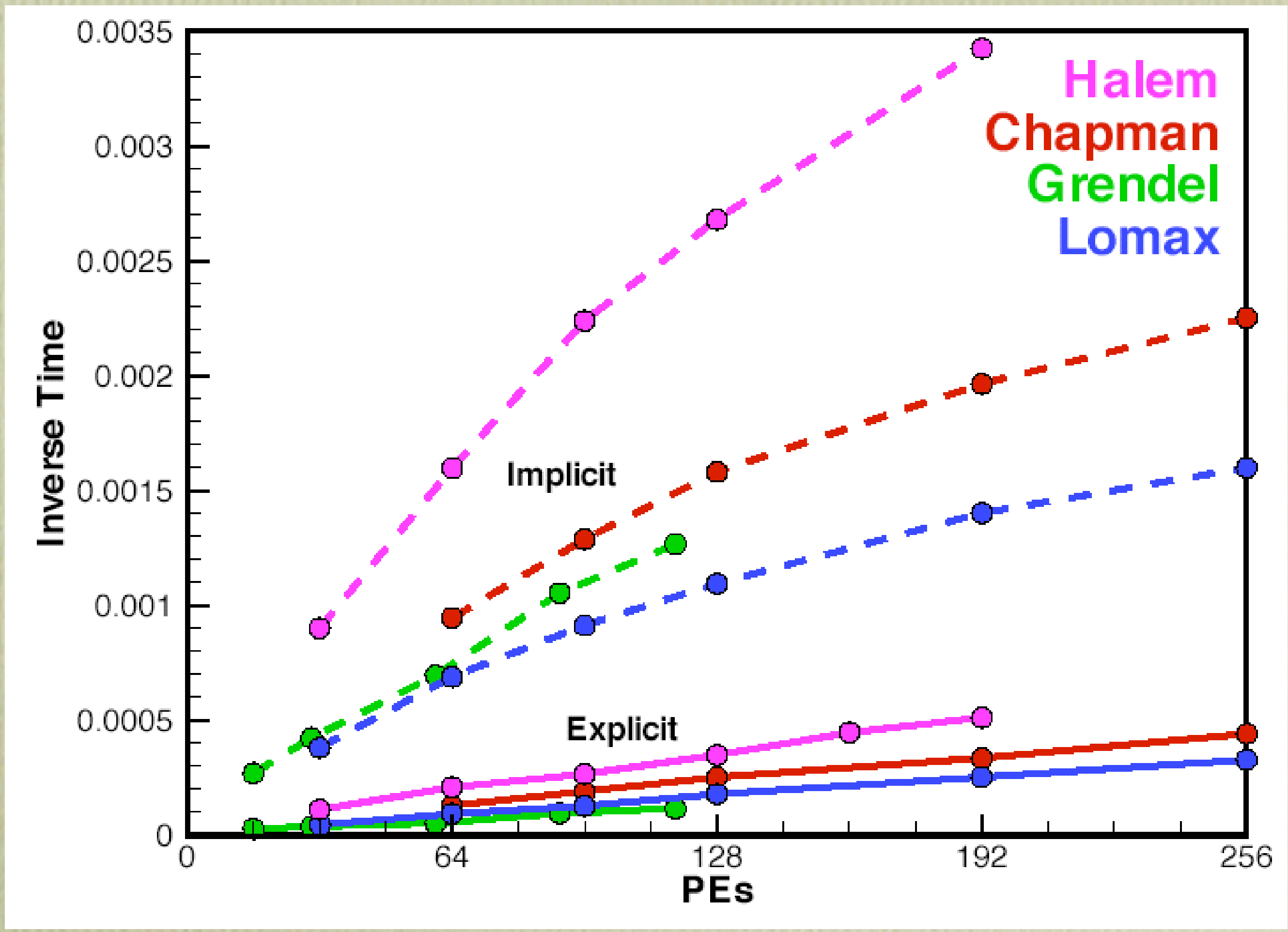
- Modified Block Incomplete Lower-Upper (MBILU) preconditioner restricted to a block:

$$A = (I - \Delta t_n \beta \partial \mathbf{R} / \partial \mathbf{U}) \approx \mathcal{L} \cdot \mathcal{U}$$

  no fill-in is allowed in L and U.

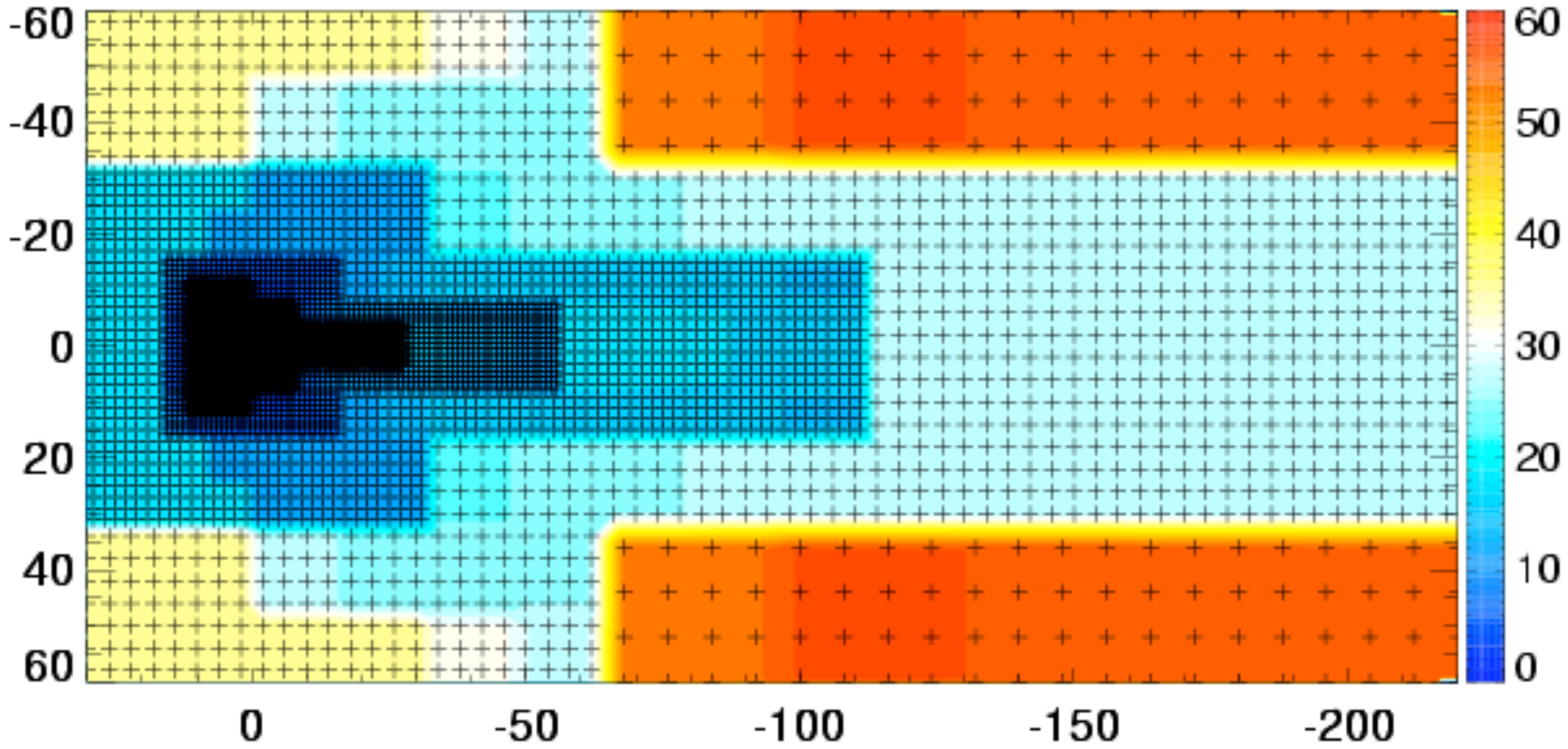
- The Jacobian used for the preconditioner is based on the first order local Lax-Friedrichs scheme and it is evaluated with numerical derivatives of flux and source functions.

Timing Results



- Halem = 192 CPU Compaq ES-45
- Chapman = 256 CPU SGI 3800
- Lomax = 256 CPU Compaq ES-45
- Grendel = 118 CPU PC Cluster (1.6 GHz AMD)

Maximum Explicit Time Step in a Magnetosphere Simulation



Explicit/Implicit Scheme

- Fully implicit scheme has no CFL limit, but each iteration is expensive (memory and CPU)
- Fully explicit is inexpensive for one iteration, but CFL limit may mean a very small Δt
- Set optimal Δt limited by accuracy requirement:

  Solve blocks with unrestrictive CFL explicitly

  Solve blocks with restrictive CFL implicitly

  Load balance explicit and implicit blocks separately

Explicit/Implicit Algorithm

1. Set time step based on accuracy, efficiency and robustness requirements
2. Assign blocks to be explicit or implicit based on local stability conditions.
3. Load balance explicit and implicit blocks separately.
4. Advance explicit blocks with one time step.
5. Update ghost cells for implicit blocks.
6. Advance implicit blocks with one time step.
7. Update all ghost cells.

Explicit/Implicit Algorithm Cont.

- Optimal time step:
  We select the optimal time step based on a few runs. One could design an adaptive algorithm.
- Order of accuracy:
  2nd order accuracy requires that the explicit blocks get the time centered flux from the implicit neighbors. Solution: apply the explicit step on all blocks then overwrite the solution in the implicit blocks.
- Conservative properties:
  It is possible to make the fluxes through the explicit/implicit interface perfectly conservative, but it requires substantial development. In our tests and applications the results are OK (as good as the conservative) without correcting the fluxes.

Time Step Control

- Non-linear instabilities are a fact of life:
  The time step has to be adjusted for sake of robustness and efficiency.

- Stability indicator:
  An MHD code typically fails with negative pressure and/or density. Define $Q = \min(p_{n+1}/p_n, \rho_{n+1}/\rho_n)$ where the minimum is taken for all grid cells.

- Time step adjustment:
  If $Q < 0.3$ then redo the time step with
  $\Delta t'_n = \Delta t_n/2$
  If $0.3 < Q < 0.6$ then reduce the next time step to
  $\Delta t'_{n+1} = 0.9\,\Delta t_n$
  If $Q > 0.8$ then increase the next time step to
  $\Delta t'_{n+1} = \min(\Delta t_{max}, 1.05\Delta t_n)$

Controlling the Divergence of B

- **Projection Scheme** (Brackbill and Barnes)
  Expensive on a block adaptive parallel grid. It may be more efficient but less robust for the implicit scheme.
- **8-Wave Scheme** (Powell and Roe)
  Simple and robust but div B is not small. Non-conservative terms. Works fine for implicit scheme, it actually improves the convergence of the Krylov solver.
- **Diffusive Control** (Dedner et al.)
  Simple but it may diffuse the solution too. Only the operator split implementation works well for the implicit scheme.
- **Constrained Transport** (Balsara, Dai, Ryu, Tóth)
  Exact but complicated. Does not allow local time stepping. Generalization to implicit scheme would be rather complicated.

MHD Code: BATSRUS

- *Block Adaptive Tree Solar-wind Roe Upwind Scheme*
- Conservative finite-volume discretization
- Shock-capturing Total Variation Diminishing schemes
- Parallel block-adaptive grid (Cartesian and generalized)
- Explicit and implicit time stepping
- Classical and semi-relativistic MHD equations
- Multi-species chemistry
- Splitting the magnetic field into $B_0 + B_1$
- Various methods to control the divergence of B

Numerical Tests

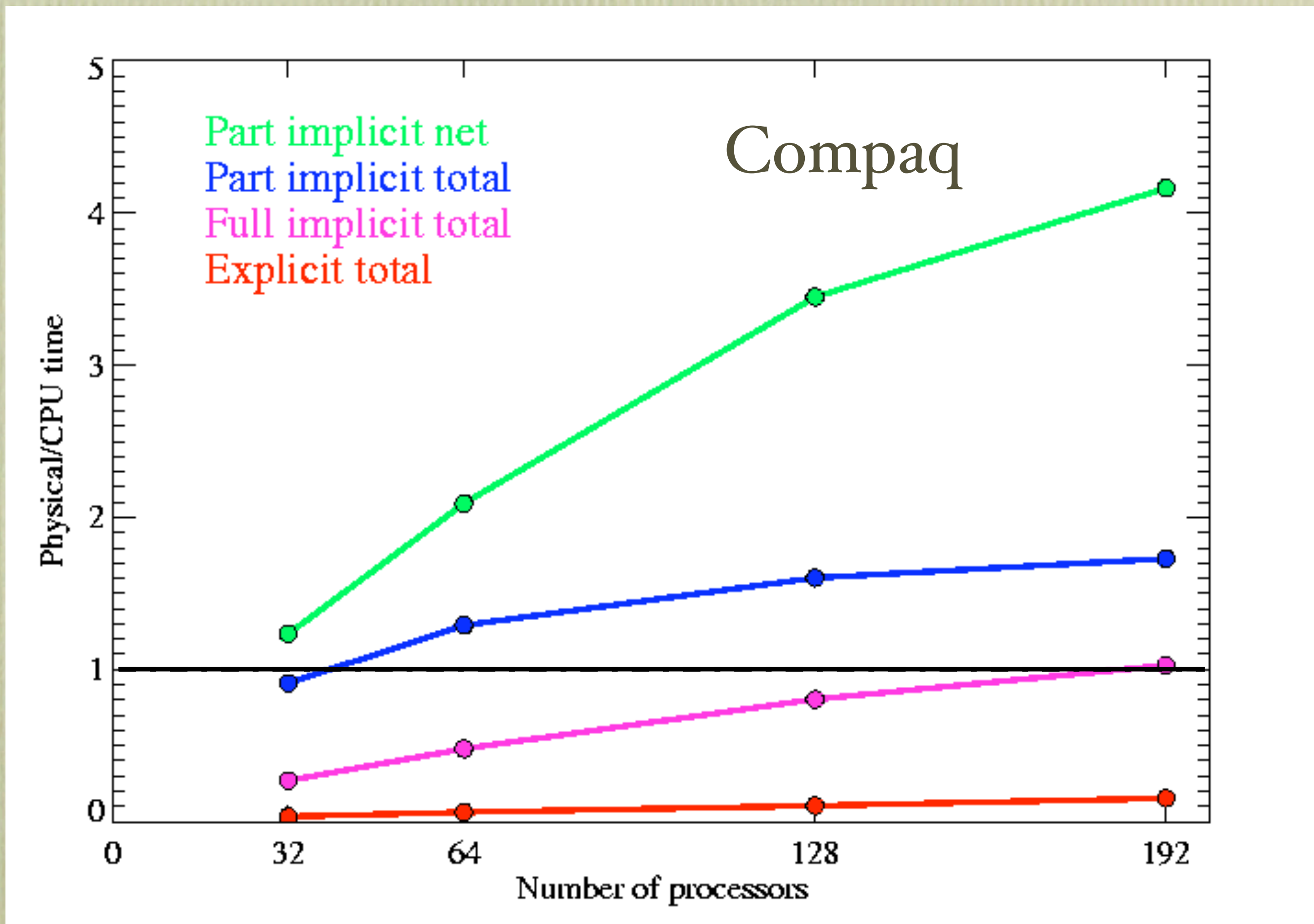
- Propagation of smooth waves:
  2nd order accuracy is demonstrated.

- Interaction of a sound wave with a magnetic discontinuity:
  Robustness, accuracy and efficiency are demonstrated.
  Choices made in the NKS solver are carefully examined.

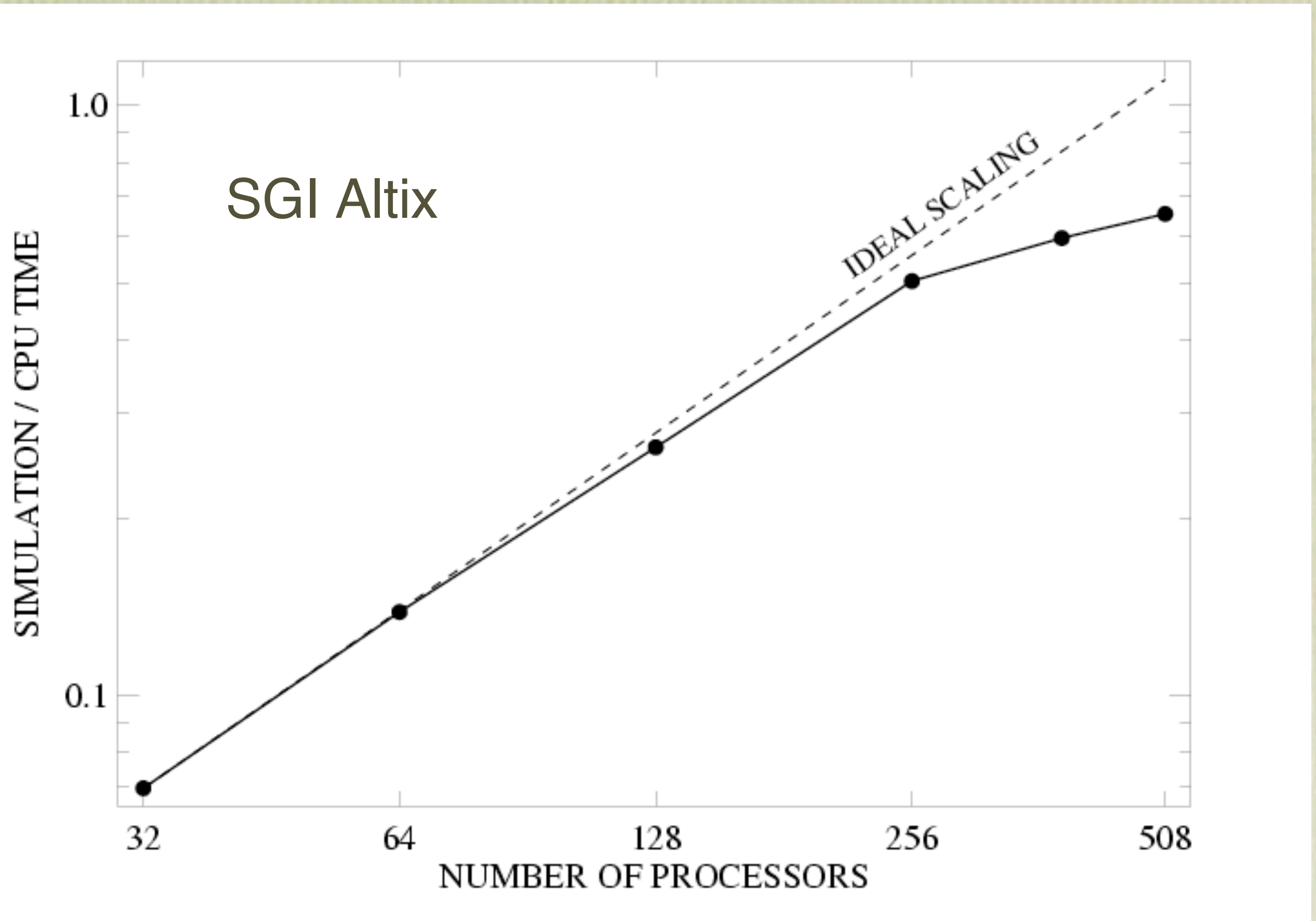
- Magnetospheric applications:
  Parallel scaling, scaling with problem size, robustness, accuracy and efficiency for space physics applications are demonstrated.

- See Toth et al. [2006, JCP in press] for more detail.

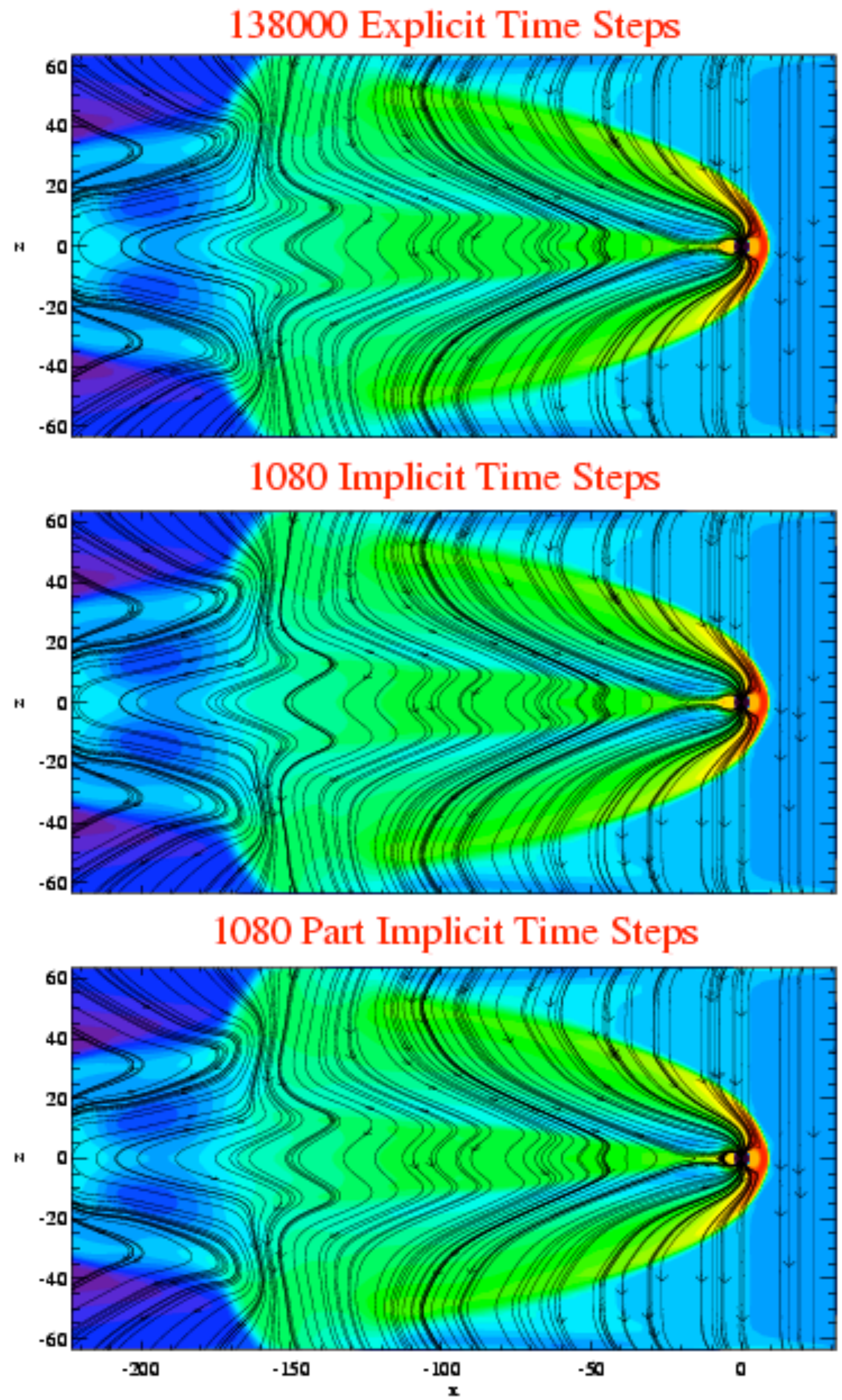
Timing Results for a Space Physics Application



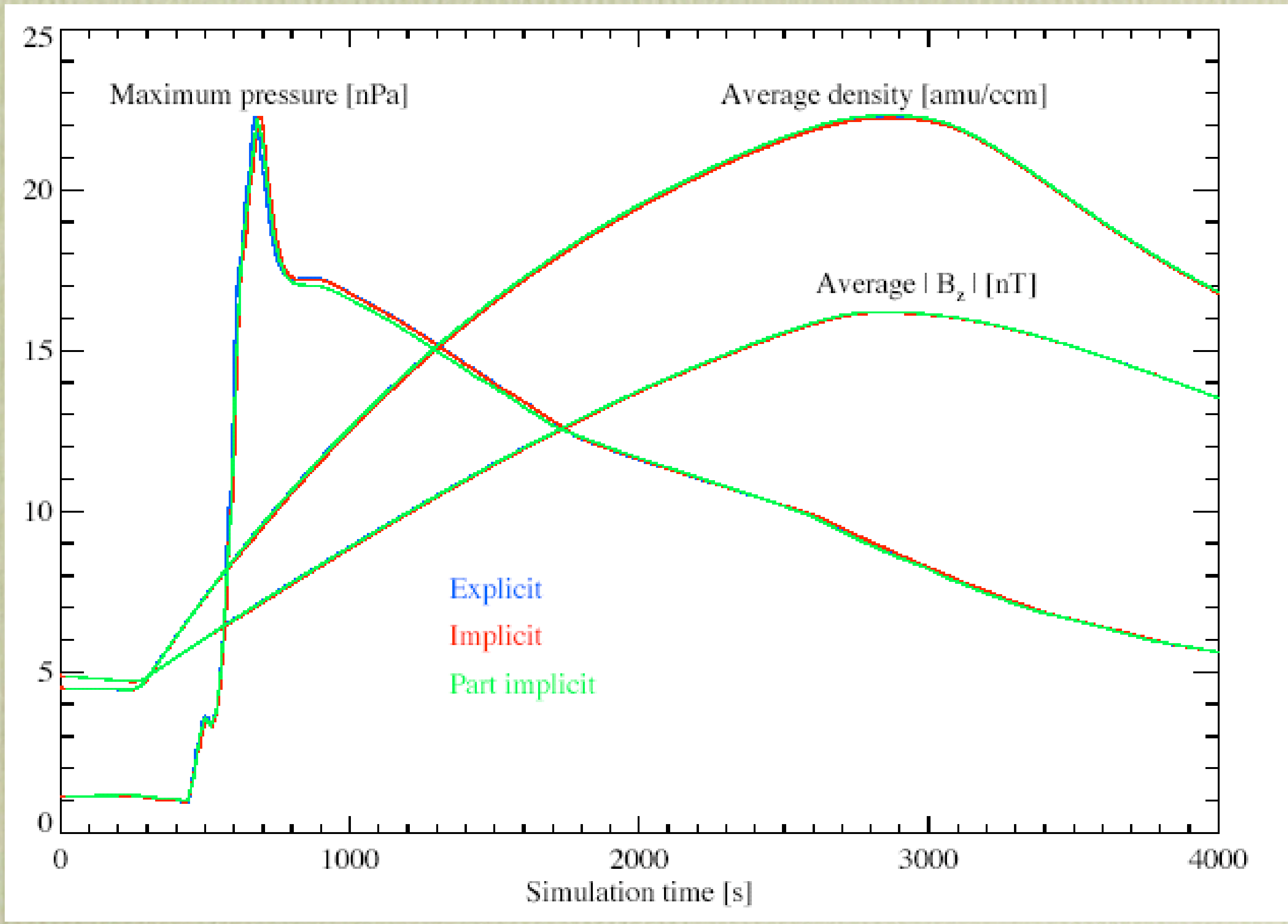
Expl./Impl. Timings for High Resolution Grid (2.3 million cells)



Comparison of results at a given time in a magnetosphere simulation



Comparison of Time Evolution of Some Average Quantities



Concluding Remarks

- The optimal choices for the Jacobian-free NKS scheme strongly depend on the application.
- The explicit/implicit scheme can give additional speed up with relatively little investment.
- We have achieved faster than real time simulation of the magnetosphere with the explicit/implicit method.